

Évaluation et mise en place des pratiques standards
à la division des Ressources informationnelles de
l'arrondissement de Saint-Laurent

par

Michel Hervé NORCEIDE

RAPPORT DE PROJET D'INTERVENTION EN ENTREPRISE PRÉSENTÉ
À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE COMME EXIGENCE
PARTIELLE À L'OBTENTION DE LA MAÎTRISE EN GÉNIE LOGICIEL

MONTREAL, LE 20 AVRIL 2020

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

©Tous droits réservés

Cette licence signifie qu'il est interdit de reproduire, d'enregistrer ou de diffuser en tout ou en partie, le présent document. Le lecteur qui désire imprimer ou conserver sur un autre média une partie importante de ce document, doit obligatoirement en demander l'autorisation à l'auteur.

PRÉSENTATION DU JURY

CE RAPPORT DE PROJET A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

Alain April, directeur de projet
Département de Génie logiciel et TI à l'École de technologie supérieure

Abdelaoued Gherbi, président du jury
Département de Génie logiciel et TI à l'École de technologie supérieure

REMERCIEMENTS

Je désire remercier certaines personnes qui m'ont aidé à concrétiser ce projet :

Le professeur Alain April qui m'a accompagné dans la réalisation de ce projet.

M. Daniel Simon et M. Philippe Couvas pour la confiance qui m'ont accordé en m'offrant la possibilité de réaliser ce projet en entreprise.

Et aussi toutes ces personnes qui m'ont apporté beaucoup d'information pour l'enrichissement de cette étude : les professeurs Claude Laporte, Sylvie Trudel et M. Marc Taillefer.

Je remercie également quelques collègues à la maîtrise qui ont participé à ce projet : Alexandre Dassonville et Chorouk Benzaki.

Aux membres de ma famille : Wislène, Mitchell et Whitney, pour leur support inconditionnel.

Évaluation et mise en place des pratiques standards à la division des Ressources informationnelles de l'arrondissement de Saint-Laurent

Michel Hervé NORCEIDE

RÉSUMÉ

Ce projet d'intervention en entreprise porte sur l'évaluation et la mise en place de pratiques standards à la section de développement logiciel de la division informatique d'une petite organisation municipale. À la suite de retards excessifs dans la livraison et des problèmes de maintenance de certaines applications, la direction souhaiterait améliorer le processus en place.

Pour atteindre cet objectif, différentes normes et pratiques standards dans le domaine des TI, ont été passés en revue. Deux de ces standards auront retenu l'attention : les méthodes agiles qui sont très populaires actuellement et la norme ISO 29110 conçue spécifiquement pour des petites organisations, de taille comparable à celle de l'équipe étudiée.

L'approche de l'amélioration continue adoptée pour ce projet recommande de dresser un bilan initial de la situation dans l'organisation avant de faire des changements. C'est l'objectif de ce présent rapport.

Dans ce bilan, la capacité d'adoption de l'agilité de l'équipe a été évaluée. Ensuite les processus en cours dans cette organisation ont été analysés et comparés aux processus recommandés par la norme ISO 29110. Enfin, les membres de l'équipe ont partagé leur point de vue par rapport aux points forts à protéger et aux faiblesses à améliorer. De ces constats et observations découlent les neuf (9) recommandations émises au chapitre 5 de ce rapport.

Mots-clés: Agile, amélioration des processus logiciels, gestion de projet logiciel, pratiques standards de génie logiciel, ISO 29110.

Assessment and implementation of standard practices in the IT division of the borough of Saint-Laurent

Michel Hervé NORCEIDE

ABSTRACT

This business intervention project focuses on the evaluation and implementation of standard practices in the development team of the IT division of a small municipal organization. Following some excessive delivery delays and maintenance issues for some applications, Management would like to improve the processes in place.

To achieve this goal, various standards of best practices in the IT field were reviewed. Two of these standards were adopted: the very popular agile methods and the ISO 29110 standard designed specifically for small organizations like the team analysed.

The continuous improvement approach used for this project recommends making an initial assessment of the organization's situation before making any changes. That's the purpose of this paper.

In this report, the team's ability to adopt agility was assessed. Then the practices in use in this organization were analyzed and compared to the ones recommended by the ISO 29110 standard. Finally, the team members shared their point of view on the strengths and weaknesses of the current process. From these observations were made nine (9) recommendations presented in the chapter 5 of this report.

Keywords: Agile, software process improvement, software projects management, software engineering best practices, ISO 29110.

TABLE DES MATIÈRES

INTRODUCTION	1
CHAPITRE 1 CONTEXTE ET PROBLÉMATIQUE	3
1.1 Contexte de l'intervention.....	3
1.2 Problématique	4
1.3 Objectifs.....	5
1.4 Portée	6
CHAPITRE 2 REVUE LITTÉRAIRE.....	7
2.1 Les référentiels : Normes et bonnes pratiques	7
2.1.1 Les normes ISO 9001 et ISO 20000	8
2.1.2 COBIT.....	9
2.1.3 CMMI	10
2.1.4 ITIL	11
2.1.5 Le modèle S3M.....	12
2.1.6 La norme ISO/IEC 29110	13
2.2 Les approches agiles	14
2.3 Les méthodes d'évaluation	18
2.3.1 La norme ISO/IEC 33001 :2015	19
2.3.2 SCAMPI.....	19
2.3.3 TIPA.....	19
2.3.4 Les trousse de déploiement ISO 29110 /ISO 20000	20
2.4 Mise en place de pratiques standards dans les petites organisations	20
2.4.2 Compatibilité entre référentiels.....	22
2.4.2.1 ISO 29110 et SCRUM	22
2.4.2.2 ISO/IEC 29110 for Service Delivery vs ITIL	23
2.4.2.3 ISO 29110 et ISO 20000.....	24
2.4.3 Conclusion	24
CHAPITRE 3 MÉTHODOLOGIE	27
3.1 Phase 1- Démarrage	28
3.1.1 Vision, buts et objectifs.....	28
3.1.2 Exigences et normes	28
3.1.2.1 Justification du choix d'ISO/IEC 29110.....	29
3.1.2.2 Les trousse de déploiement	29
3.1.2.3 Les outils d'évaluation.....	30
3.2 Évaluation initiale	30
3.2.1 Évaluation de la capacité d'adoption de l'agilité.....	31
3.2.2 Évaluation des processus de développement logiciel	31
3.3 Phase 2 - Mise en place des recommandations	32
3.4 Phase 3 - Évaluation et Rapport final	32
3.5 Budget et ressources nécessaires	32
3.6 Calendrier.....	33

CHAPITRE 4 EVALUATION	34
4.1 Évaluation de l'Agilité.....	34
4.2 Évaluation des processus de Génie logiciel	35
4.2.1 Description du processus en place	36
4.2.2 Évaluation du processus en place par rapport au processus ISO 29110 ...	38
CHAPITRE 5 RECOMMANDATIONS	40
5.1 Adoption de l'agilité	40
5.1.1 Travailler en équipe	40
5.1.2 Favoriser l'excellence technique par la formation continue	41
5.1.3 Instaurer les réunions de rétrospectives	41
5.2 Mise en place des pratiques standards	42
5.2.1 Planification et suivi des projets	42
5.2.2 Mise en place d'un gestionnaire de code source.....	42
5.2.3 Mise en place des pratiques de tests	43
5.2.4 Documenter la conception, le code et la configuration.....	43
5.3 Soutien organisationnel.....	44
5.3.1 Gestion du changement.....	44
5.3.2 Appui de la direction.....	44
CONCLUSION.....	45
ANNEXE I Guide d'entrevue	47
ANNEXE II Grille d'évaluation du processus d'Implémentation du logiciel.....	49
ANNEXE III Gabarit du plan projet actuellement utilisé.....	51
ANNEXE IV Grille d'observations	53
BIBLIOGRAPHIE	56

LISTE DES TABLEAUX

	Page
Tableau 1. Les 4 valeurs du Manifeste Agile (Agile Alliance, 2001)	15
Tableau 2. Principales différences entre Scrum et Kanban (Atlassian).....	17
Tableau 3. Critères de sélection des modèles et normes.....	22
Tableau 4 – Les trousse de déploiement pour les TPO (<i>Tiré de Laporte</i>)[11].....	30
Tableau 5- Tableau méthodologique	33
Tableau 6- Processus d’implémentation du logiciel de la DRI.....	37

LISTE DES FIGURES

	Page
Figure 1: Organigramme de la Division des Ressources informationnelles.....	4
Figure 2. Représentation étagée de CMMI-DEV (tiré de Basque (2011))	11
Figure 3. Comparaison des méthodes agiles : XP, Scrum et Kanban.....	18
Figure 4. Grille d'évaluation de mise en œuvre du profil basique (tiré de [11])	20
Figure 5- Les 6 étapes de l'amélioration continue – (<i>Adapté de Tremblay (2020) [4]</i>)	27
Figure 6 - Application des principes agiles dans les projets.....	34
Figure 7- Activités du profil d'entrée. (Source: site web du prof. Claude Laporte).....	36
Figure 8 - Évaluation du processus de gestion de projet (PM).....	38
Figure 9 - Évaluation du processus d'implémentation du logiciel	39

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ASL :	Arrondissement de Saint-Laurent
CMMI :	Capability Maturity Model Integration
DRI :	Division des Ressources Informationnelles
DRI-Dev :	Section Développement applicatif de la DRI
DRI-Ops :	Section Opérations et infrastructure de la DRI
DSA:	Direction des Services administratifs
ÉTS :	École de Technologie Supérieure
IEC :	International Electrotechnical Commission
ISO :	International Organization for Standardization
ITIL :	Information Technology Infrastructure Library
ITSM :	Information Technology System Management (Système de gestion TI)
SPI :	Software process improvement
TPO :	Très petites organisations (moins de 25 employés)
TI :	Technologies de l'Information

INTRODUCTION

Ce projet de recherche appliquée de 15 crédits s'inscrit dans le cadre du programme de maîtrise en génie logiciel. Ce projet est réalisé dans le contexte d'une organisation municipale dont la direction a décidé d'améliorer l'efficacité de l'équipe et la qualité des applications et des services offerts par sa division informatique.

Ce projet vise, dans un premier temps, l'évaluation des processus en cours actuellement à la division informatique et l'identification de recommandations pour l'amélioration de son niveau de maturité. Dans une deuxième phase, la planification et la mise en place des pratiques standards de développement logiciel.

La norme ISO 29110 sera le référentiel de pratiques exemplaires choisi pour ce projet. Toutefois d'autres types de pratiques éprouvées, telles que celles utilisées dans les développements agiles seront proposées afin de maximiser les chances d'atteindre les objectifs fixés (c.-à-d. l'amélioration des pratiques en cours) par la direction de l'organisation.

Ce rapport d'évaluation est composé de cinq chapitres. Le premier chapitre décrit le contexte de l'organisation et la problématique. Le chapitre 2 passe en revue quelques normes et pratiques standards actuellement utilisées en technologie de l'information. Le chapitre 3 décrit la méthodologie utilisée. Les résultats de l'évaluation sont analysés au chapitre 4 et les recommandations qui découlent de ces constats sont présentées au dernier chapitre.

CHAPITRE 1

CONTEXTE ET PROBLÉMATIQUE

1.1 Contexte de l'intervention

L'Arrondissement de Saint-Laurent (ASL) est une municipalité de la Ville de Montréal. C'est un des rares arrondissements qui dispose de sa propre équipe informatique indépendante du Service des technologies de l'information (STI) de la Ville de Montréal. Cette division a été maintenue à l'arrondissement après la fusion des municipalités en 2002 parce qu'elle offrait des applications « maison » et un service de proximité que les services centraux n'étaient pas en mesure d'offrir.

La DRI relève de la Direction des services administratifs (DSA). Elle est divisée en deux sections. La section Développement applicatif et Support aux Systèmes de Gestion (DEV) développe et maintient plus d'une trentaine d'applications utilisées à l'interne par les 300 employés de l'organisation, ou à l'externe par les citoyens. Certaines de ces applications sont utilisées par d'autres arrondissements ayant des besoins identiques. La section Infrastructure et Opération (OPS) gère le parc informatique et l'infrastructure réseautique dans les différents édifices et supporte les utilisateurs.

L'équipe informatique est composée d'une quinzaine de personnes: cinq (5) développeurs, six (6) techniciens bureautiques, deux (2) administrateurs de réseau, un chef de section, sous la responsabilité du chef de division des Ressources informationnelles (DRI).

L'organigramme présenté à la figure 1 situe la Division des Ressources informationnelles dans la structure générale de l'arrondissement.

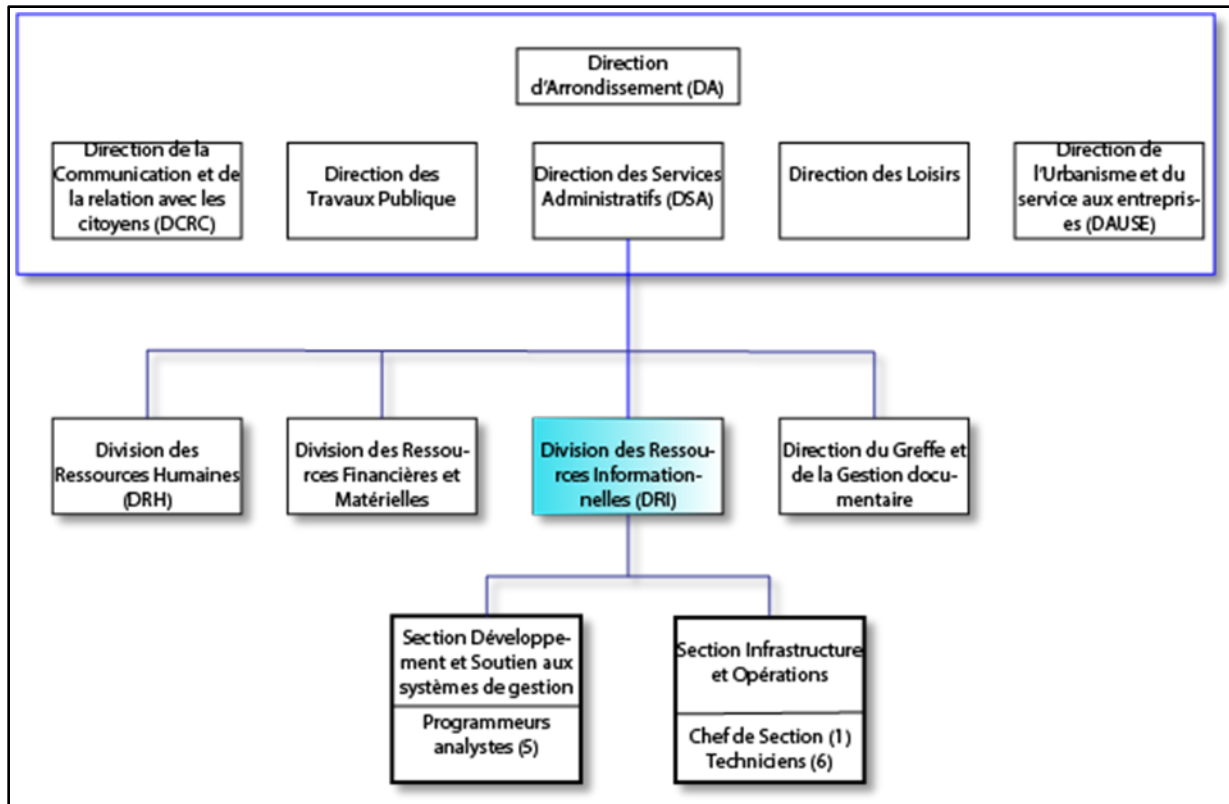


Figure 1: Organigramme de la Division des Ressources informationnelles

1.2 Problématique

Au cours de la dernière année, il y a eu beaucoup de changements au niveau de la division informatique, à la suite du départ de plusieurs employés. Ces événements ont provoqué une remise en question au niveau de la direction :

- Est-il encore justifié d'avoir un service informatique indépendant, pendant que les autres arrondissements ont intégré le Service TI corporatif de la VDM?
- Quels seront les avantages et inconvénients à maintenir le service en place?
- Qu'est-ce qui devrait être fait pour maintenir l'infrastructure et le service à un niveau optimal?
- Quel serait le coût en termes de ressources matérielles et humaines pour maintenir ce service en opération ?

La Direction de l'arrondissement estime qu'il est primordial de maintenir en place la division informatique. Conséquemment, il serait important de s'assurer d'offrir un service de qualité.

En tant qu'employé de l'organisation, depuis une dizaine d'années, et candidat à la maîtrise en génie logiciel à l'École de technologie supérieure, j'ai identifié une opportunité d'amélioration du service en introduisant les pratiques normalisées. Le standard retenu servira de référence pour évaluer et faire des propositions d'amélioration à la direction et, subséquemment à l'acceptation de ces recommandations, mettre en place et faire le suivi des améliorations observées.

1.3 Objectifs

1.3.1 Objectifs de la Direction

Il a été décrit que l'objectif général de la direction est de renforcer les capacités de l'équipe informatique à fournir un service de qualité, tout en utilisant les pratiques exemplaires reconnues par l'industrie. De manière plus spécifique, la direction aimerait :

- Réduire les délais de traitement des demandes de services;
- Réduire les délais de mise en production des applications développées;
- S'assurer que le service et les applications répondent adéquatement aux besoins des utilisateurs;
- Garantir la maintenance et la mise à jour des applications.

1.3.2 Hypothèse et objectifs du projet d'intervention

L'hypothèse de départ de ce projet est que l'application soutenue des pratiques normalisées aurait comme effet, à moyen terme, une augmentation mesurable du niveau de maturité de la DRI et permettrait par conséquent d'atteindre les objectifs de la direction. Les objectifs de ce projet de recherche appliquée sont :

- D'évaluer les processus en place à la Division des Ressources Informationnelles (DRI) par rapport à un standard reconnu;
- D'émettre des recommandations concernant des pratiques à mettre place en vue d'élever le niveau de maturité de la division informatique.

1.4 Portée

Cette évaluation initiale analysera vise uniquement la section « Développement applicatif » de la DRI. Cette étude se limitera à analyser les processus et pratiques en cours par rapport aux standards de l'industrie.

L'évaluation des processus de « Gestion de services TI » serait recommandée dans le but d'améliorer parallèlement les processus en cours dans la section « Opérations et infrastructure ». Toutefois certaines contraintes empêchent la réalisation de ce volet à court terme. Des études plus techniques en termes d'architecture réseau seraient aussi utiles pour achever ce rapport afin d'offrir un portrait global et complet aux décideurs.

Bien que la DSI soit quelquefois impliquée dans des processus d'acquisition de logiciel, ce volet ne sera pas pris en compte dans le présent projet. Ce domaine de processus est règlementé par des directives corporatives qui ne sont pas du ressort de la DSI.

CHAPITRE 2

REVUE LITTÉRAIRE

Pour s'améliorer, il est primordial de pouvoir se mesurer, c'est-à-dire d'avoir un référentiel par rapport auquel on pourra se comparer.

Il existe différents normes et référentiels de pratiques exemplaires dans le domaine de l'ingénierie informatique. Afin de s'assurer que le modèle choisi répondra au besoin de l'organisation, il est important de passer en revue la liste des référentiels les plus utilisés. Il serait aussi utile d'étudier comment d'autres organisations de taille comparable ont réussi leur démarche d'amélioration de processus, quels outils elles ont utilisés, et surtout, apprendre de leurs erreurs.

Dans cette optique, quelques ouvrages de référence et une vingtaine d'articles scientifiques (incluant des rapports de projet d'entreprise) ont été consultés. Les mots-clés soumis aux moteurs de recherche étaient les suivants: norme et bonnes pratiques en TI; amélioration de processus de développement logiciel; CMMI; ITIL; SCAMPI et méthodes agiles. Un des critères de sélection des articles était qu'ils soient assez récents (c.-à-d. moins de 10 ans) afin de cibler les plus récentes versions des différents modèles.

2.1 Les référentiels : Normes et bonnes pratiques

Une norme est un ensemble de règles de lignes directrices établies par consensus, approuvée par un organisme reconnu et faisant l'objet d'une publication. Une norme a pour but de garantir que des produits ou services ont un certain niveau de qualité. Une norme a souvent un caractère obligatoire.

Une bonne pratique est une technique ou une méthode appliquée avec succès par plusieurs organisations qui est reconnue comme telle par l'industrie. Ce sont des cadres de travail qui font des recommandations pour accéder aux meilleures caractéristiques d'un produit ou aux meilleures pratiques d'un métier¹.

Certaines de ces normes sont généralistes telles qu'ISO 9001 ou le PMBOK et peuvent s'appliquer à différents types d'organisation et de projets. D'autres sont plutôt techniques tels qu'ISO 20000, COBIT (Control Objectives for Information and related Technology), CMMI (Capability Maturity Model of Integration) ou ITIL (Information Technology Infrastructure Library). Certains de ces modèles s'appliquent seulement à une partie de l'organisation (Dufay, 2010, p11).

Ces normes et modèles de processus logiciels sont importants pour le développement de logiciels (et la fourniture de service) parce que leur mise en œuvre a généré des avantages rapportés dans la littérature, tels que: la réduction des coûts, le respect du calendrier, meilleure qualité et meilleure satisfaction des clients [15].

La section suivante décrit quelques normes et référentiels parmi les plus populaires actuellement utilisés. Une courte description du référentiel sera présentée et dans certains cas un bref retour d'expérience des professionnels du domaine. La plupart des définitions sont tirées de l'ouvrage de Baud (2016), chapitre 1 [1].

2.1.1 Les normes ISO 9001 et ISO 20000

La norme ISO 9001 fait partie d'un ensemble de norme lié à la gestion de la qualité, quel que soit le secteur d'activité. La version 2008 préconise une approche par processus, complétée par une exigence d'amélioration continue. Elle est basée sur huit (8) principes: orientation client,

¹ Définitions tirées des notes de cours MTI825 – Gestion des Services TI, Daniel Tremblay, ETS, Hiver 2020

leadership, implication du personnel, approche par processus, gestion par approche système, amélioration continue, la prise de décision sur des indicateurs factuels, les relations mutuelles entre les parties prenantes.

La Norme ISO 20000 est spécialisée dans la gestion des services informatiques. Elle s'appuie sur une approche par processus basé de la version 2 d'ITIL. Elle définit, mais aussi planifie, la gestion des services informatiques. Elle nécessite la mise en œuvre de 13 processus, regroupés en 5 domaines:

- La fourniture des services;
- La gestion des relations;
- La résolution;
- Le contrôle;
- La mise en production.

La certification ISO 20000 nécessite un grand effort de mise en œuvre. Selon Dufay (2010), les normes internationales ISO 9001 et ISO 20000 sont de haut niveau et ne rentrent pas dans le détail d'implémentation des pratiques. Il est alors conseillé de s'appuyer sur un modèle technique plus détaillé pour aider à la mise en place des pratiques.

2.1.2 COBIT

Le référentiel COBIT (Control Objectives for Information and Related Technologies) vise la gouvernance des TI. C'est une démarche qui permet l'audit et l'évaluation des services informatiques d'une entreprise afin d'en évaluer la performance et la robustesse en termes de sécurité et de conformité. C'est un référentiel complet qui permet de contrôler l'ensemble des opérations liées aux systèmes d'information [1].

Créé dans les années 1990 par l'association américaine ISACA (Information Systems Audit and Control Association), COBIT est utilisé comme outils de conformité dans de nombreuses normes et standards. Au fil des versions successives, il tend à devenir un outil fédérateur de gouvernance des systèmes d'information en intégrant progressivement les apports d'autres référentiels tels qu'ISO 9000 et ITIL.

COBIT est un référentiel basé sur une approche par processus, ce qui permet une complémentarité avec ITIL v3. La version 5 de COBIT disponible depuis 2012 définit 37 processus regroupés en cinq domaines. La dernière version COBIT 2019 a été publiée vers la fin de 2018.

2.1.3 CMMI

Le modèle CMMI permet d'évaluer le niveau de maturité des processus d'ingénierie informatique et en particulier les activités de développement applicatif. L'idée de base est que la qualité d'un produit est déterminée par la qualité du processus employé pour le concevoir, le réaliser et le maintenir. Il a été créé dans les années 1980 par la SEI (Software Engineering Institute) du Carnegie Mellon University pour mesurer la maturité des processus logiciels des fournisseurs informatiques pour le compte du département de la défense américaine.

Le CMMI, dans sa version 1.3, tente de couvrir différents domaines de processus par le biais de ses trois constellations : CMMI-DEV pour le développement, CMMI-SVC pour la gestion des services, CMMI-ACQ pour l'acquisition de systèmes ou de produits. La dernière version du modèle CMMI v2.0 a été publiée en 2018.

Le CMMI définit une échelle de maturité des processus à 5 niveaux, ainsi que des indicateurs pour évaluer les activités pour atteindre ces niveaux. La version 1.3 définit 22 domaines de processus (Process Area PA) pour CMMI-DEV, répartis sur les 5 niveaux de maturité: initial, discipliné, défini, maîtrisé, mesuré. (Figure 2)

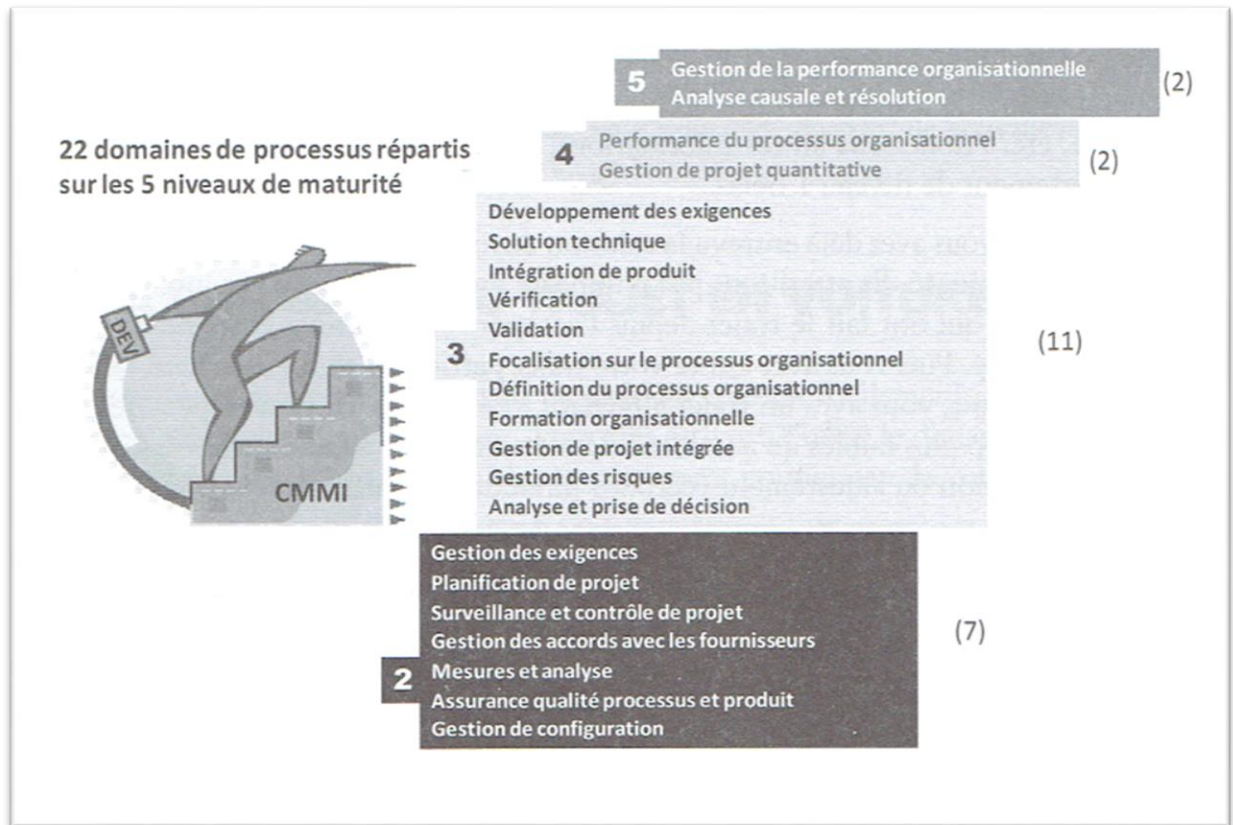


Figure 2. Représentation étagée de CMMI-DEV (tiré de Basque (2011))

2.1.4 ITIL

Le référentiel ITIL est une sélection de pratiques exemplaires opérationnelles en matière de gestion des services informatiques. Il propose un cadre de travail, des recommandations et la norme ISO 20000 permet de certifier ses processus. ITIL suggère une approche globale de « comment » gérer les services informatiques de bout en bout qui s'articule autour d'un cycle de vie des services informatiques en cinq phases: la stratégie, la conception, la transition, l'exploitation du service et l'amélioration continue.

Les guides ITIL ont été créés en Angleterre à la fin des années 80 et publiés par l'OGC (Office of Government of Commerce) du ministère de Commerce britannique. ITIL est à la base de la norme ISO 20000.

En février 2019, AXELOS, l'organisme britannique chargé de gérer le référentiel de bonnes pratiques ITIL, a annoncé la publication de la version 4. C'est une refonte en profondeur qui met l'accent sur la chaîne de valeur soutenue par des principes et des pratiques au lieu d'un cycle de vie basé sur des processus. Cette nouvelle édition du référentiel prend en compte les grandes tendances technologiques et méthodologiques des dernières années : développement de l'intelligence artificielle, adoption du « cloud » et les approches agiles.

Un composant clé du nouveau référentiel est la chaîne de valeur du service (SVS). Ce concept décrit comment les composants et activités au sein d'une organisation fonctionnent ensemble pour créer de la valeur, à la fois pour les organisations, leurs clients et l'ensemble des parties prenantes (Tremblay, 2020) [4].

2.1.5 Le modèle S3M

Le Modèle d'évolution de la capacité de la maintenance du logiciel (S3M) est un modèle d'amélioration des processus de maintenance du logiciel. Il vise une approche d'amélioration qui offre aux organisations les éléments indispensables pour l'optimisation des processus de la maintenance de logiciel (April et Abran, 2008) [5]. Il regroupe les activités uniques de la maintenance logicielle tout en conservant une structure comparable à celle du modèle de maturité CMMI™ (CMMI-Product-Team, 2010, cité par Marchand 2018, p.40).

Selon Wikipédia, la version récente du S3M — Version 2 — a été lancée en janvier 2006.

Le modèle d'évaluation de la maturité de maintenance S3m est divisé en quatre domaines de processus contenant 18 processus clés.

S3M se concentre sur les petites activités de la maintenance. Autrement dit, pour les processus d'un logiciel opérationnel qui évolue grâce à une nouvelle fonctionnalité ou la correction d'erreurs. Si cette mise en œuvre requiert la composition d'une équipe, elle pourra être gérée comme un projet de développement. Ce projet de maintenance fait évoluer l'application logicielle comme le cas d'un nouveau projet de développement. Les activités de la maintenance regroupent toutes les autres opérations de correction et d'amélioration du logiciel (April et Abran, 2008).

2.1.6 La norme ISO/IEC 29110

La norme ISO/CEI 29110 concerne l'ingénierie de systèmes et l'ingénierie du logiciel, les processus de cycle de vie pour les très petites organisations (TPO) du domaine du logiciel. Le terme TPO se réfère à une entreprise, une organisation ou un département [informatique] ayant moins de 25 personnes² [11].

La norme internationale ISO/IEC 29110 a été publiée en 2011 pour aider à augmenter la capacité et la compétitivité de ces petites organisations qui œuvre dans le domaine du logiciel par l'amélioration des processus et, par conséquent, des produits et /ou services fournis. Dans l'introduction de la norme (ISO/IEC 29110,2016) il est indiqué que:

« Il a été constaté que les TPE éprouvent des difficultés à relier les Normes internationales à leurs besoins et à justifier les efforts requis pour appliquer les normes à leurs pratiques. La plupart des TPE ne peuvent ni se permettre les ressources, en termes de nombre d'employés, d'expertise, de budget et de temps, ni voir un avantage net dans la mise en place de systèmes trop complexes ou de processus de cycle de vie logiciel. » [Traduction libre]

La norme ISO/IEC 29110-2 présente le concept de profils standardisés pour les TPO. Un ensemble de 4 profils y est décrit:

² Les informations sur la Norme ISO/IEC29110 proviennent du site web du Prof. Claude Laporte.

- Le profil d'entrée est destiné aux TPO qui travaillent sur des petits projets (c.-à-d. 6 mois-personnes) et des startups;
- Le profil basique cible les TPO développant une seule application par une seule équipe de travail;
- Le profil intermédiaire est destiné aux TPO développant plus d'un projet en parallèle avec plus d'une équipe de travail;
- Le profil avancé est destiné aux TPE qui souhaitent se maintenir et se développer en tant qu'entreprise indépendante de développement de systèmes et/ou de logiciels.

Le profil basique du génie logiciel est composé de deux processus :

- Le processus de gestion de projet;
- Le processus de mise en œuvre du logiciel.

Chaque processus est composé d'un ensemble d'activités et chaque activité est composée d'un ensemble de tâches.

La partie 5-3 de la norme (ISO/IEC TR 29110-5-3:2018) trace les lignes directrices de prestation des services pour les TPO. Elle est inspirée de la Norme ISO 20000. Ce guide traite de 4 processus liés à la gestion des services :

- Processus de gouvernance (GO);
- Processus de contrôle des services (CO);
- Processus des relations de service (RE);
- Processus d'incident de service (IN).

Cette norme met aussi à la disposition des petites entreprises des trousse de déploiement (de l'anglais *Deployment Packages*). Ces trousse proposent des références vers de la documentation libre d'accès, vers des outils, des gabarits, des listes de vérification ainsi que du matériel de formation.

2.2 Les approches agiles

Habituellement, les méthodes agiles sont classées dans la catégorie des normes internes de développement (c.-à-d. cycle de vie de développement logiciel). Il est généralement accepté que l'utilisation de méthodes agiles apporte un avantage compétitif pour les organisations. Bustard et Al (2013) mentionne que: « *Optimal software engineering means creating the right*

product, through the right process, to the overall satisfaction of everyone involved. Adopting the agile approach to software development appears to have helped many companies make substantial progress towards that goal » [15]. Par conséquent on ne peut pas penser améliorer la qualité du service et des processus d'un département œuvrant dans le domaine du logiciel en laissant de côté l'évaluation de l'agilité.

L'agilité est « une philosophie de travail basée sur des valeurs et des principes. Les méthodes et approches dites « Agiles » sont cohérentes avec ces valeurs et ces principes. » (Boisvert et Trudel (2011), chapitre 1). Elle fournit des méthodologies de travail pour des équipes auto organisées.

Les valeurs et les principes de l'agilité ont été définis en 2001 par un groupe de 17 experts en développement logiciel. Leur objectif était de trouver une meilleure approche pour développer des logiciels qui pourraient apporter de la valeur au client en réduisant les délais. Ils ont synthétisé leur réflexion dans le Manifeste agile. Celui-ci définit quatre valeurs (tableau 2.1) et 12 principes.

Tableau 1. Les 4 valeurs du Manifeste Agile (Agile Alliance, 2001)

Nous découvrons comment mieux développer des logiciels par la pratique et en aidant les autres à le faire. Ces expériences nous ont amenés à valoriser :
1. Les individus et leurs interactions plus que les processus et les outils. 2. Des logiciels opérationnels plus qu'une documentation exhaustive. 3. La collaboration avec les clients plus que la négociation contractuelle. 4. L'adaptation au changement plus que le suivi d'un plan.
Nous reconnaissons la valeur des seconds éléments, mais privilégions les premiers.

Parmi les nombreuses méthodes agiles existantes, celles qui sont actuellement les plus utilisées dans les petites équipes sont: Scrum, Extreme Programming (XP) et Kanban, selon le « *13th Annual State of Agile Report* » VersionOne (2019).

2.2.1. Scrum

Scrum est un cadre empirique de gestion de projet agile, utilisé pour fournir des incréments de grande valeur au client, de manière itérative. Scrum s'appuie sur des équipes auto organisées. Il s'appuie également sur un représentant du client, le « Product Owner », qui fournit à l'équipe une liste des fonctionnalités souhaitées. Ces fonctionnalités sont priorisées en fonction de la valeur qu'elles apportent au client. Cette méthode a été présentée pour la première fois à la conférence OOPSLA³ par Ken Schwaber en 1995.

Scrum est idéal pour les projets où le client n'est pas tout à fait sûr des fonctionnalités à développer. Le travail à effectuer est répertorié dans le carnet de produit (Product Backlog). Le projet est réalisé par itérations (c.-à-d. un Sprint) qui ont une durée généralement de deux à quatre semaines.

Scrum définit 3 rôles: le « Scrum master », le « Product Owner » et l'équipe de développement. Au début de chaque itération, une réunion de planification de Sprint est organisée au cours de laquelle le « Product Owner » présente les éléments prioritaires du carnet de produit et l'équipe Scrum sélectionne les tâches à effectuer lors du « Sprint » à venir. Ces tâches sont ensuite déplacées du « Product Backlog » vers le « Sprint Backlog ». Chaque jour pendant le sprint, l'équipe organise une brève réunion de synchronisation appelée « Daily Scrum ». À la fin de chaque « Sprint », l'équipe démontre la fonctionnalité terminée lors d'une réunion de revue de sprint.

2.2.2. Kanban

Selon Wikipédia, la méthode Kanban est « un système visuel de gestion des processus qui indique quoi produire, quand le produire et en quelle quantité. Cette approche est directement

³ OOPSLA (en anglais « Object-Oriented Programming, Systems, Languages & Applications ») est une conférence annuelle organisée aux États-Unis.

inspirée du système de production de Toyota et des méthodes Lean ». Cette méthode applique l'amélioration continue et vise l'élimination du gaspillage.

La méthode Kanban est la moins restrictive de toutes les méthodes agiles. Elle n'impose pas ou ne prescrit aucun rôle, comme le responsable de produit ou le « Scrum Master ». Elle n'impose pas de rencontres, comme la mêlée quotidienne ou la rétrospective n'exige pas de former des équipes multidisciplinaires et elle s'adapte à la participation des spécialistes. (Marchand,2018). Le tableau 2 présente les principales différences entre Scrum et Kanban selon le site Atlassian.

Tableau 2. Principales différences entre Scrum et Kanban (Atlassian)

	Scrum	Kanban
Cadence	Regular fixed length sprints (ie, 2 weeks)	Continuous flow
Release methodology	At the end of each sprint	Continuous delivery
Roles	Product owner, scrum master, development team	No required roles
Key metrics	Velocity	Lead time, cycle time, WIP
Change philosophy	Teams should not make changes during the sprint.	Change can happen at any time

2.2.3. XP (eXtreme Programming)

Selon le site agilealliance.org, la programmation extrême (XP) est un cadre de développement logiciel agile qui vise à produire des logiciels de meilleure qualité. XP est le plus technique des cadres agiles en ce qui a trait aux pratiques d'ingénierie pour le développement de logiciels. Elle est née officiellement en octobre 1999 avec le livre « *Extreme Programming Explained* » de Kent Beck. Don Wells, sur le site extremeprogramming.org, décrit les contextes où l'utilisation de XP est appropriée :

- Exigences logicielles en évolution dynamique;
- Risques causés par les projets à temps fixe utilisant les nouvelles technologies;
- Petite équipe de développement étendue colocalisée;
- La technologie utilisée permet des tests unitaires et fonctionnels automatisés.

La figure 3 montre une comparaison des trois méthodes selon leur niveau de rigueur.

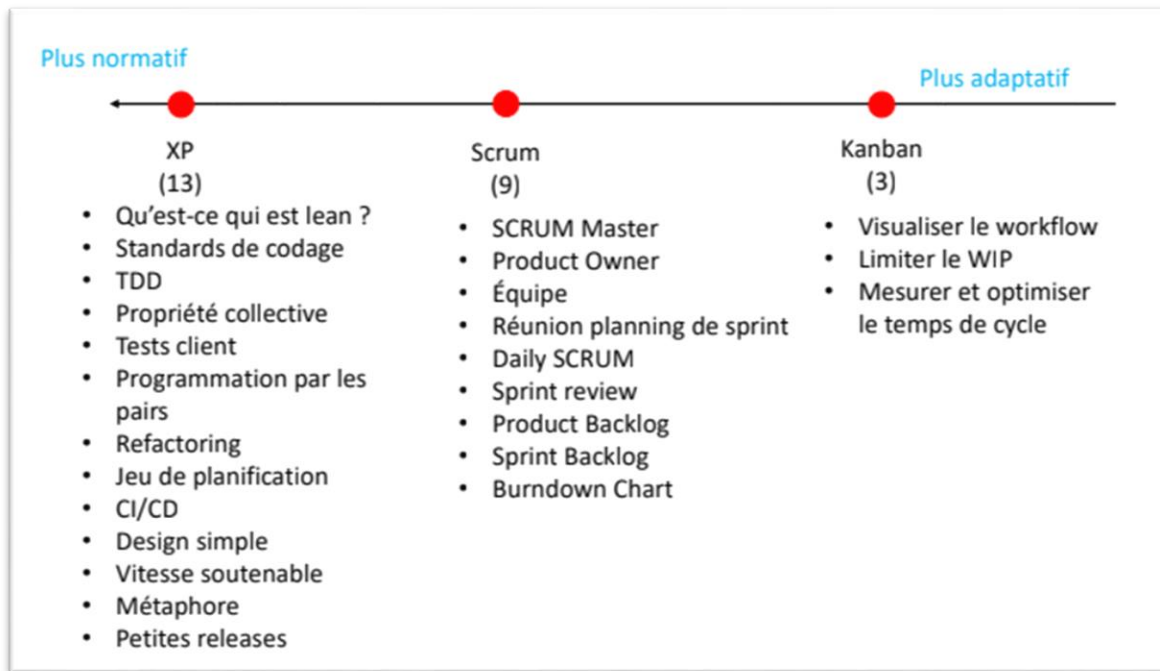


Figure 3. Comparaison des méthodes agiles : XP, Scrum et Kanban. (Tiré de Dassonville et Al (2019)) [23]

2.3 Les méthodes d'évaluation

Toute démarche d'amélioration de processus commence par une évaluation. La position de l'organisation est mesurée par rapport au référentiel choisi qu'il soit ISO, ITIL ou CMMI. Cette évaluation initiale permet:

- De mesurer l'écart (gap) entre la situation actuelle et souhaitée;
- D'identifier les objectifs d'améliorations, les critères de succès et les indicateurs;
- De définir une ou plusieurs actions d'amélioration et de les prioriser.

Il existe toute une panoplie de méthodes d'évaluation. Parmi celles-ci, on en a sélectionné quelques-unes qui pourraient être utiles dans le cadre de ce projet.

2.3.1 La norme ISO/IEC 33001 :2015

La norme ISO 15504 (anciennement SPICE) conçue pour l'évaluation des processus logiciels, est une norme de qualité des processus logiciels. Elle permet de déterminer les niveaux de capacité des processus. Elle a été partiellement remplacée par la norme ISO/IEC 33001: 2015.

Cette norme permet d'effectuer l'évaluation en tant qu'autoévaluation ou évaluation indépendantes (Rout, Walker et Dorling, 2017). Dans les deux cas, la norme exige que les évaluateurs soient qualifiés.

2.3.2 SCAMPI

La méthode d'évaluation SCAMPI (*Standard CMMI™ Appraisal Method for Process Improvement*) est la méthode d'évaluation préconisée par la SEI. Elle permet l'analyse approfondie des forces et faiblesses des processus de développement logiciels d'une organisation, soulève les risques de développement et détermine les niveaux de capacité et de maturité des processus. (SCAMPI-Upgrade-Team, 2011), cité par Marchand (2018).

La méthode SCAMPI comporte trois classes d'évaluation notées : A, B ou C allant de la plus stricte à la moins contraignante. Une évaluation de classe A, par exemple, requiert que l'évaluateur soit certifié par le CMMI. Une évaluation SCAMPI peut nécessiter beaucoup de ressources et est relativement coûteuse. Dufay [2] précise que l'un des intérêts majeurs de la méthode SCAMPI est de permettre l'enregistrement auprès du SEI. Sinon on peut utiliser une méthode d'évaluation propre ou inspirée de SCAMPI.

2.3.3 TIPA

L'évaluation des processus ITSM de Tudor (TIPA) est une méthodologie qui fournit un cadre et des étapes pour effectuer une évaluation du niveau de capacité des processus de gestion des services informatiques. TIPA est conforme à l'approche normalisée d'ISO/IEC 15504-2 pour effectuer une évaluation de la conformité des processus de gestion des services informatiques.

Cette approche peut être utilisée par les organisations impliquées dans la planification, la gestion, le suivi, le contrôle et améliorer l'acquisition, la fourniture, le développement, l'exploitation, l'évolution et le support des produits et services. »

2.3.4 Les trousse de déploiement ISO 29110 /ISO 20000

Quelques outils ont été développés par des étudiants de l'ÉTS pour faciliter la mise en place de l'amélioration des processus dans les TPO. Entre autres, on y trouve la « *Grille d'évaluation de mise en œuvre du profil basique* » (Figure 2.2) (System_engineering_ISO_29110_Basic_profile_Activity_Assessment.xls), la « *trousse pour la conduite d'un audit interne* » (Deployment Package Conduct of Internal Audit) (Kabli, 2009).

System Engineering - ISO 29110 Basic Profile								
Project Management	ID	Task List	Input Products	Output Products	Roles	Is implemented	ID	
PM.1 Project Planning	PM.1.1	Review the <i>Statement of Work</i>	Statement of Work	Statement of Work [reviewed]	PJM SYS	Partially		
	PM.1.2	Define with the <i>Acquirer</i> the <i>Delivery Instructions</i> of each one of the <i>Deliverables</i> specified in the <i>Statement of Work</i> .	Statement of Work [reviewed]	Project Plan • Delivery Instructions	PJM ACQ	Partially		
	PM.1.3	PM.1.3 Define the System Breakdown Structure (SBS) that represents the relationship between the system and its system elements. Note: the system boundaries must be defined Note: this task is iterative as the SBS is based on the System Design Document (SDD). The SDD is at the beginning preliminary and all system elements hierarchy is not necessary defined completely. The SBS is updated while the SDD is progressively completed.	System Design Document	Project Plan • System Breakdown Structure	PJM DES	Partially		
	PM.1.4	Select a product lifecycle and define milestones according to the <i>Statement of Work</i>	Project Plan • System Breakdown Structure Statement of Work	Project Plan • Milestones	PJM WT	Partially		
	PM.1.5	Identify the specific <i>Tasks</i> to be performed in order to produce the <i>Deliverables</i> and	Statement of Work [reviewed]	Project Plan • Tasks	PJM			

Figure 4. Grille d'évaluation de mise en œuvre du profil basique (tiré de [11])

2.4 Mise en place de pratiques standards dans les petites organisations

Les caractéristiques les plus importantes de ce projet concernent la taille de l'organisation et son périmètre multisectoriel qui inclut des projets de développement logiciel et des services de support TI.

Laporte et coll. (2014) insiste sur le fait qu'il est très difficile pour les petites organisations de mettre en place une démarche d'amélioration de processus basé sur les normes reconnues telles que CMMI. « *Small software organizations found that the processes described in software standards are too complicated, have no guides for easy implementation, are not easy to integrate, and require additional resources not usually available in business* ».

Cette section analysera comment les petites organisations ont réussi la mise en place de leur démarche d'amélioration de processus, quels sont les référentiels et les méthodes utilisées et les défis auxquels ils ont eu à faire face.

2.4.1. Les critères de sélection du référentiel

Le tableau 3, inspiré de Dufay (2010), résume les caractéristiques des différents référentiels analysés. À partir de ce tableau on peut constater qu'il n'existe pas un référentiel qui couvre complètement et de manière optimale, tous les besoins d'une organisation du domaine du logiciel. Mais, puisque la norme ISO 29110 se positionne comme le standard pour les petites organisations du domaine du logiciel, il serait avantageux de considérer la possibilité de l'utiliser avec d'autres référentiels compatibles et plus aptes à couvrir des besoins spécifiques non couverts par cette norme.

Tableau 3. Critères de sélection des modèles et normes

Critères \ Référentiels	ISO 9001	ISO 20000	COBIT	ITIL	CMMI Dev	CMMI Svc	CMMI Acq	S3M	ISO/IEC 291100
Périmètre de haut niveau: Organisation – gouvernance	R	-	R	-	-	-	-		
Périmètre multisectoriel	R	R	-	-	-	R	R		
Périmètre Service informatique	-	R	-	R	-	R	-		?
Périmètre projet	-	-	-	-	R	-	-		R
Périmètre sous-traitance	-	R	-	-	-	-	R		?
Maintenance								R	
Petite et moyenne organisation (PMO)									R
Modèles d'évaluation / Audit disponibles	R	R	R	E	R	R	R		R
Certification de l'organisation	R	R	E	-	R	R	R		R
Certification des personnes	-	-	R	R	R	R	-		
Légende : R: Recommandé. Forte adhérence avec le critère de sélection E: envisageable. Adhérence possible moyennant adaptation -: Pas d'adhérence spécifique à souligner ?: Aucune information									

2.4.2 Compatibilité entre référentiels

2.4.2.1 ISO 29110 et SCRUM

Plusieurs projets ont été réalisés dans différents pays sur l'utilisation d'ISO 29110 pour l'amélioration des processus dans les petites entreprises. De nombreux auteurs ont publié des cas de réussite dans la mise en œuvre de cette norme. Tous ont démontré de très bons résultats

dans la mise en œuvre de l'ISO/CEI 29110 dans les petites organisations, comme la réduction des reprises, l'accès à de nouveaux clients, l'augmentation de la qualité, et l'amélioration de leurs processus entre autres (O'Connor R.V. et Laporte C.Y., 2012).

Une étude réalisée en 2015 a fait une analyse de conformité des méthodologies agiles à l'aide de la norme ISO/IEC 29110. Les chercheurs ont conclu que « la méthode SCRUM présente un niveau de conformité élevé avec le processus de gestion de projet d'ISO/IEC 29110, tandis que la méthode XP a un niveau de conformité plus modéré » [13].

Une autre étude réalisée en 2018 [8] décrit l'expérience de mise en œuvre de l'ISO/IEC 29110 dans un groupe de 4 entreprises de développement logiciel au Mexique. Ces petites entreprises utilisent la méthodologie SCRUM en combinaison avec les pratiques du profil standard et ont été certifiées par les auditeurs de NYCE, l'organisme de certification mexicain. Cette étude démontre que l'application de la norme ISO/CEI 29110 permet de renforcer les pratiques de SCRUM dans les TPO du domaine du logiciel.

2.4.2.2 ISO/IEC 29110 for Service Delivery vs ITIL

Sur la page web de iso.org, il est mentionné qu'il existe une version d'ISO 29110 pour la gestion des services (ISO/IEC 29110 for Service Delivery), toutefois il existe très peu d'étude sur l'implantation de ce modèle en entreprise. Par contre, ITIL a été largement adopté partout dans le monde pour la gestion des services informatiques. La question maintenant est de savoir : est-ce qu'ITIL s'implante bien dans les petites organisations?

Une revue de littérature systématique réalisée par Melendez et al (2015), sur les modèles de gestion des services de technologie de l'information appliqués aux moyennes et petites organisations a montré que le référentiel le plus populaire est la version 2011 d'ITIL (85.7%) des entreprises, suivi de CMMI-Svc (2009) et COBIT (1999). Selon la même étude, les

processus les plus utilisés sont: la gestion des incidents (47.1%), la gestion des accords de niveau de service (29.4%) et la gestion des problèmes (29.4%).

Dans une autre étude réalisée à Madrid en 2015, les chercheurs ont mené deux enquêtes sur la mise en place d'ITIL dans les PME, dans la région de Madrid d'une part et dans plusieurs pays autour du monde d'autre part. Ces enquêtes ont révélé que l'approche qui se concentre sur la gestion des incidents et des problèmes (Service Desk) est celle qui est adoptée en premier dans la majorité des entreprises, car ces processus permettent d'obtenir plus rapidement des résultats parce qu'ils sont orientés vers le client.

2.4.2.3 ISO 29110 et ISO 20000

En 2014, une étudiante de L'ÉTS a réussi à mettre en place un ITSM (Système de gestion TI) dans une petite organisation gouvernementale en utilisant les trousse de déploiement adaptées de la norme ISO 20000. Ces trousse ont été développées dans un projet antérieur (Kabli, 2009). « les trousse sont conçues de telle sorte qu'une TPO peut mettre en œuvre son contenu sans avoir à mettre en œuvre le référentiel complet de la norme ISO/CEI 20000 » (Dufour, 2014). Dans sa démarche, elle a identifié 9 des 13 processus d'ISO 20000 qui apportaient le plus de valeur pour l'organisation.

Il existe une certaine relation entre les normes ISO 20000 et ISO 29110. Les trousse de déploiement créées pour faciliter la mise en place dans les TPO, ont une structure identique. Utiliser ces outils pour la mise en place des bonnes pratiques dans les deux sections de la DSI pourrait être un avantage.

2.4.3 Conclusion

À partir de plusieurs expériences de mise en œuvre d'une démarche d'amélioration de processus dans les petites organisations (TPO) du domaine du logiciel, plusieurs chercheurs ont compris que le succès de la démarche ne dépend pas seulement des atouts du référentiel ou de la norme. Il faut aussi prendre en compte la résistance au changement, la culture

organisationnelle, la motivation et la volonté des équipes de travail, le soutien inconditionnel de la direction et une communication efficace [10]. « *Cet ensemble de facteurs indique que le succès de l'application d'un modèle d'amélioration de processus, dans les PME, est lié à l'engagement des parties prenantes, au-delà du choix du modèle approprié, il faut que les personnes s'engagent dans ce changement* » (Coque-Villegas et al, 2016). [Traduction libre]

CHAPITRE 3

MÉTHODOLOGIE

L'approche choisie pour ce projet est basée sur le principe de l'amélioration continue, représenté par la figure 5. Les six étapes de l'amélioration continue sont réparties sur les trois phases du projet: la phase de démarrage, la phase d'implantation et la phase de rédaction du rapport final.

Cette section décrit les grandes étapes du projet et présente les outils et les ressources qui seront utilisés.

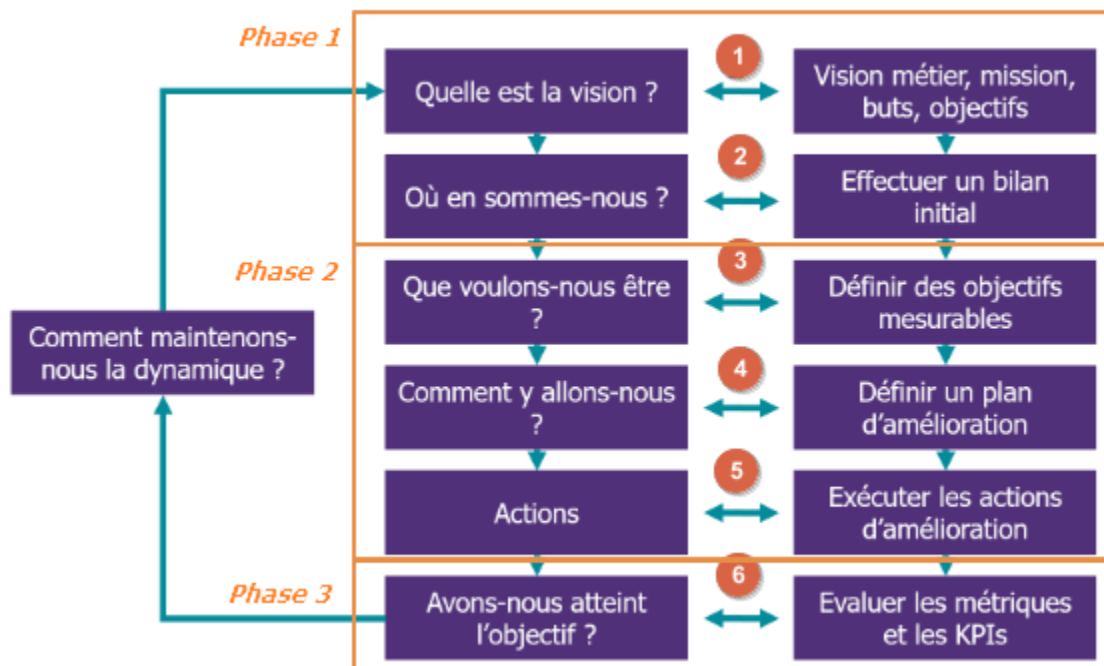


Figure 5- Les 6 étapes de l'amélioration continue – (Adapté de Tremblay (2020) [4])

3.1 Phase 1- Démarrage

La première phase du projet, objet de ce présent rapport, regroupe les deux (2) premières étapes du processus d'amélioration continue. L'objectif de cette phase est de:

- 1) préciser la vision et les exigences du projet;
- 2) réaliser l'évaluation initiale.

3.1.1 Vision, buts et objectifs

a) Vision

Au cours des rencontres mensuelles de mobilisation des employés de la DRI, il a été défini que la direction de ASL voudrait faire de la DRI un pôle de service informatique efficace, répondant aux besoins des employés et des citoyens, reconnu et accepté par le Service des Technologies de l'Information corporatif (STI).

b) But :

Il a été convenu avec la direction que le but de ce projet n'est pas l'obtention d'une certification. L'équipe aimerait tout simplement améliorer progressivement ses pratiques par la standardisation des processus.

c) Objectifs

Les objectifs du projet ont été définis au chapitre 1.

3.1.2 Exigences et normes

La liste des exigences du projet sera établie en fonction de la norme choisie comme référence. Bien que l'objectif ne soit pas l'obtention d'une certification, la direction reconnaît que l'application des normes peut favoriser l'harmonisation des processus et rendre le service plus efficace. Dans ce contexte, la norme ISO 29110, a été choisie comme cadre de référence afin de structurer l'intervention.

3.1.2.1 Justification du choix d'ISO/IEC 29110

La norme ISO/IEC 29110 a été retenue, car elle est actuellement promue dans de nombreux pays à travers le monde. Elle est moins complexe que les autres normes, ce qui la rend plus adaptée aux petites organisations. De plus, cette norme n'impose pas une approche de développement, elle peut facilement s'adapter aux méthodes agiles. L'expérience d'implantation de cette norme dans une institution financière canadienne, relatée par Plante et Laporte (2018), est très concluante: « *The implementation of the new process ISO/IEC 29110-agile process has resulted in significant reduction in the number and the impact of incidents.* »[22]

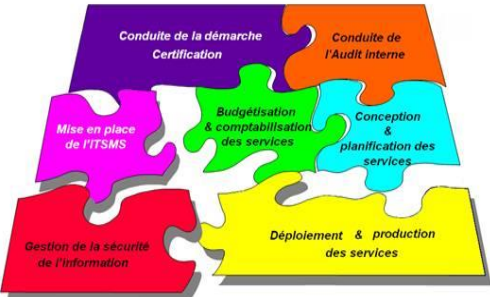
Selon les critères décrits dans le guide TR 29110-5-1-2, l'organisation correspond très bien au profil d'entrée de la trousse de déploiement : (c.-à-d. *une petite organisation développant des petits projets ne dépassant pas 6 mois-personnes*). C'est cette trousse qui sera utilisée pour la gestion des projets de développement logiciel.

L'autre avantage de cette norme c'est que le guide technique TR 29110-5-1-3, sur la prestation des services a sélectionné 4 des 13 processus d'ISO 20000 (c.-à-d. ITIL) qui seraient adaptés aux petites organisations. Par conséquent, si dans l'avenir la DSI décide de migrer vers ITIL, qui est le standard de l'industrie, il sera très facile d'utiliser les processus mis en place avec cette norme.

3.1.2.2 Les trousse de déploiement

Les trousse de déploiement (*Deployment Package -DP*) ont été développées pour faciliter la mise en place des normes ISO 29110 et ISO 20000, dans les TPO. Ces trousse serviront d'échafaudage pour aider à redresser les processus existants, ou pour mettre en place de nouveaux processus qui seraient intéressants pour l'organisation.

Tableau 4 – Les trousse de déploiement pour les TPO (*Tiré de Laporte*)[11]

ISO 29110 - Ingénierie logicielle	Prestation des services
 LES TROUSSES DE DÉPLOIEMENT DU PROFIL BASIQUE	Trousse de déploiement pour l'implémentation de la gestion des services dans les TPO 

3.1.2.3 Les outils d'évaluation

Les outils d'autoévaluation inclus dans les trousse de déploiement seront utilisés dans ce projet. Les principaux avantages de ces outils sont :

- 1) Ils se présentent sous la forme de fichier Excel facile à comprendre et à utiliser;
- 2) Les résultats s'affichent automatiquement sous forme de diagramme;
- 3) Ces trousse développées en collaboration avec l'ÉTS ont été gracieusement fournies par les responsables. [11]

3.2 Évaluation initiale

L'évaluation initiale de la DRI comprend deux activités principales:

- 1) L'évaluation de la capacité d'adoption de l'agilité de l'équipe de développement;
- 2) L'évaluation des processus de développement logiciel en place par rapport aux recommandations de la norme ISO 29110.

3.2.1 Évaluation de la capacité d'adoption de l'agilité

La direction a manifesté un vif intérêt pour l'adoption des méthodes agiles. Mais tel qu'il a été relaté dans différentes études, le succès d'une démarche d'amélioration des processus dépend en grande partie de l'implication de l'équipe. Il est donc important dès le départ d'évaluer la capacité de l'équipe à adopter l'agilité. Cette activité comprend les tâches suivantes:

- 1) Rencontre de lancement pour présenter les objectifs du projet et les principes et les valeurs du Manifeste agile à l'équipe, incluant les gestionnaires;
- 2) Distribution d'un formulaire d'évaluation basé sur les 12 principes de l'agilité, pour évaluer dans quelle mesure chacun des développeurs est ouvert à l'application de ces principes dans leurs projets;
- 3) Entrevue avec trois personnes-ressources de la division : Directeur, Gestionnaire de projets et le développeur principal.

Les questions abordées durant les entrevues peuvent être consultées à l'Annexe-1 (Guide d'entrevue).

3.2.2 Évaluation des processus de développement logiciel

L'objectif de cette activité est d'évaluer le processus de développement utilisé à la DRI et de dresser une liste de recommandations basée sur le guide de gestion et d'ingénierie du profil d'entrée ISO/IEC 29110

L'absence de processus documenté ne signifie pas forcément l'inexistence de processus. Dans ce sens la démarche consistera à:

- 1) Analyser les documents existants produits dans les projets antérieurs;
- 2) Réaliser des entrevues avec les développeurs pour comprendre la gestion d'un projet de développement du début à la fin;
- 3) Compiler les observations dans la grille d'évaluation afin de dresser le portrait de l'équipe.

3.3 Phase 2 - Mise en place des recommandations

À la suite de l'évaluation initiale, les recommandations seront analysées et priorisées. La mise en place peut se faire de manière itérative. Ce qui reviendrait à répéter les étapes 3- 4- 5 et 6 à chaque itération (voir Tableau 5). Les principales activités de cette phase sont les suivantes⁴:

- 1) Établir les priorités à la suite du diagnostic posé et choisir les processus à implanter;
- 2) Préparer un plan du projet d'amélioration ou de mise en place de processus;
- 3) Exécuter les actions d'amélioration;
- 4) Évaluer les résultats.

Un projet pilote sera choisi pour tester l'approche Agile-ISO 29110 tel que décrit dans Plante Laporte (2018). Les deux processus du profil d'entrée de l'ingénierie logiciels, définis dans les trousse de déploiement de la norme ISO 29000 seront implantés. Il s'agit du processus de gestion de projet et du processus d'implémentation du logiciel.

3.4 Phase 3 - Évaluation et rapport final

Dans un délai de six mois suivant le début des travaux d'implantation ou d'amélioration des processus, une évaluation utilisant la même grille d'évaluation sera réalisée afin d'évaluer le progrès réalisé. Le rapport final présentant le bilan de l'expérience sera élaboré. Ce bilan présentera les progrès réalisés, ainsi que les leçons apprises...

3.5 Budget et ressources nécessaires

Puisque ce projet est réalisé dans le cadre d'un projet de maîtrise, il n'occasionnera pas de coût exceptionnel pour l'organisation. Toutefois, l'accès à certains logiciels pourrait exiger des coûts de licence au cas où ils ne seraient pas disponibles.

⁴ Ces activités des phases 2 et 3 sont mentionnées à titre indicatif. Elles ne sont pas incluses dans cette première phase du projet. Les détails de ces activités seront précisés ultérieurement.

3.6 Calendrier

Le projet devrait s'étendre sur une période ne dépassant pas un an. (Voir tableau 5).

Tableau 5- Tableau méthodologique

Étape	Activités	Ressources	Livrables	Échéancier
Phase 1- Démarrage				
Étape 1	Définir la vision	Direction, Employés	Vision	2020-01-07
	Analyser les exigences et les normes	Articles, Rapport de projet	Revue de littérature	2020-02-17
Étape 2	Évaluation initiale			
	Planification de l'évaluation		Méthodologie	2020-03-17
	Rencontre de lancement	Gestionnaires, membre de l'équipe	Engagement des parties prenantes	
	Évaluation de l'agilité	Principes agiles Guide d'entrevue Questionnaires	Notes observations	2019-12-01
	Évaluation des processus Dev	Documents, outils sur les processus utilisés	Grille d'évaluation	2020-04-01
	Rédaction du rapport		Rapport d'évaluation	2020-05-01
Phase 2- Mise en place				
Processus développement logiciel		Équipe développement		
Étape 3	Priorisation des processus à améliorer ou à implanter	Rapport d'évaluation	Liste priorisée	2020-05-01
Étape 4	Planification du projet pilote	Project management (PM) /ISO29110	Document de projet	2020-06-01
Étape 5	Réalisation du projet	Software Implementation, (SI) / ISO29110 Product Backlog	Logiciel opérationnel	2020-10-30
Phase 3 - Évaluation et rapport final				
Étape 6	Évaluation du projet pilote	Équipe Développement	Bilan de projet	2020-11-01
	Évaluation des processus mise en place	Équipe Opérations	Bilan de projet	2020-12-01
	Rédaction du rapport final	Bilans, documents de projets...	Rapport final	2021-01-01

CHAPITRE 4

EVALUATION

Cette section présente les résultats de l'évaluation selon la méthodologie définie au chapitre précédent. Les observations pertinentes sont soulignées avec les conséquences éventuelles sur la qualité du produit, ou de la productivité de l'équipe.

4.1 Évaluation de l'Agilité

L'évaluation de l'application des principes agiles est basée sur l'opinion des membres de l'équipe de développement qui ont rempli un sondage en ligne. Dans ce sondage, les 5 développeurs ont évalué sur une échelle de 1 à 5 dans quelle mesure ils appliquent chacun des 12 principes agiles dans la réalisation de leurs projets. Les résultats sont présentés dans le graphique ci-dessous.

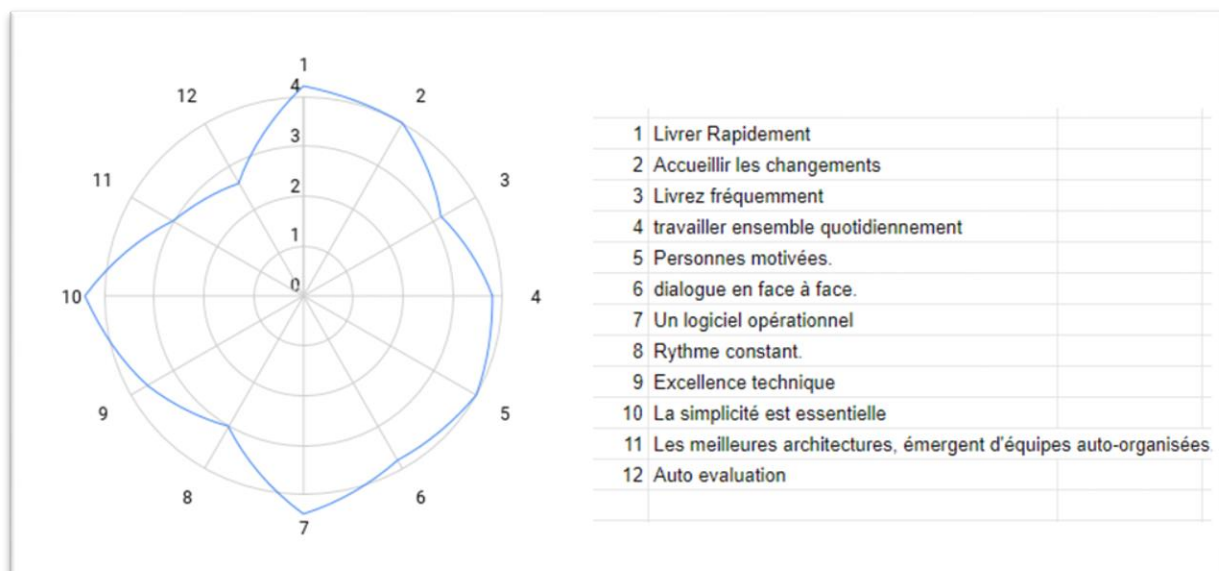


Figure 6 - Application des principes agiles dans les projets

Cette équipe n'a jamais prétendu être agile. La plupart des membres de l'équipe ignoraient même le concept d'agilité. Mais après la présentation des valeurs et des 12 principes de l'agilité, tous les membres de l'équipe ont été unanimes à reconnaître que certains valeurs et principes agiles ont été depuis toujours appliqués de façon intuitive par les membres de l'équipe. Par exemple:

- La livraison des valeurs d'affaire du client;
- La proximité avec le client pour comprendre ses besoins;
- L'acceptation des changements;
- Un logiciel fonctionnel plutôt que de la documentation exhaustive.

Cependant, certains points faibles ont été observés:

- L'incapacité à soutenir un rythme constant;
- L'impossibilité de se mettre d'accord sur une architecture unanimement acceptée;
- L'absence d'autoévaluation.

Conséquences

Les conséquences positives de cette manière de fonctionner sont:

- Les applications répondent aux besoins des clients;
- La productivité est relativement bonne.

Les conséquences négatives :

- Il y a très peu de collaboration sur les projets;
- Chacun développe comme il veut;
- Il n'y a pas d'autoévaluation donc on perd les bénéfices de l'amélioration continue.

4.2 Évaluation des processus de génie logiciel

Les processus de génie logiciel défini dans le profil d'entrée de la norme ISO 29110 sont au nombre de deux (2) : le processus de gestion de projet (PM), et le processus d'implémentation

du logiciel (SI). Les relations entre les activités de ces deux processus sont présentées à la figure 7.

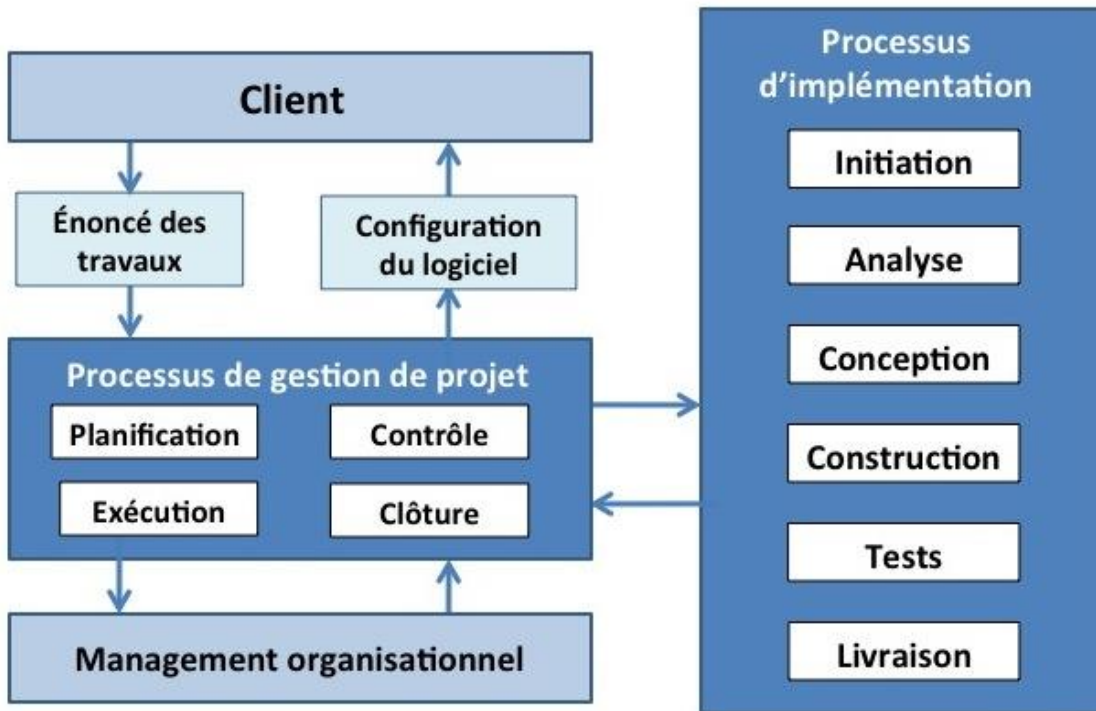


Figure 7- Activités du profil d'entrée. (Source: site web du prof. Claude Laporte)

4.2.1 Description du processus en place

Le processus en place actuellement n'est pas documenté. À partir de ce qui a été observé sur place, il pourrait ce résumé ainsi :

Quand un service de l'arrondissement (c.-à-d. le client) présente un nouveau projet de développement logiciel à la DRI, le projet fait l'objet d'une première évaluation de la part des gestionnaires respectifs pour juger de la pertinence du projet. Si le projet est accepté, une première rencontre est organisée avec le client et un analyste-programmeur de la DRI. La suite des activités s'enchaîne telle que présentée au tableau 6.

Tableau 6- Processus d'implémentation du logiciel de la DRI

No.	Activité	Objectif	Livrable	Responsable
1	Rencontre avec le client	Comprendre les besoins du client	Liste des fonctionnalités	Client CD, AP
2	Analyse et conception	Conception d'une solution	Plan de projet (minimaliste)	AP
3	Développement et présentation d'une preuve de concept au client.	Valider la compréhension du problème et de la conception.	Prototype	AP Client
4	Codage	Développement de l'application	Logiciel (version bêta)	AP
5	Tests fonctionnels	Vérification par le client, détection de bogue.	Liste de bogues Demande de changement	AP, Client
6	Correction + Ajout de fonctionnalités	Corriger les problèmes et appliquer les recommandations du client	Version stable du logiciel	AP
7	Présentation finale et formation des utilisateurs	Prise en main par le client	Logiciel fonctionnel	AP Client
8	Mise en production	Installation et configuration de l'application sur le serveur de production	Logiciel déployé	AP, AS
9	Support et maintenance	Corriger, mettre à jour, et faire évoluer l'application	Version améliorée	AP

Rôles :

- Analyste-Programmeur (AP)
- Chef de division (CD)
- Client
- Administrateur de système (AS)

Dans ce processus, on remarque qu'une seule personne, l'analyste-programmeur, accomplit toutes les tâches de l'analyse à la mise en production. À la fin du processus, il s'occupe aussi du déploiement et de la maintenance.

4.2.2 Évaluation du processus en place par rapport au processus ISO 29110

Les résultats obtenus à partir de la grille d'évaluation incluse dans la trousse de déploiement sont présentés dans les figures 8 et 9. La liste des activités et des tâches définies pour ces deux processus et leur poids respectif dans l'évaluation sont présentés à l'annexe 2.

Durant l'évaluation, il a été remarqué que le processus de gestion de projet est très faiblement implémenté du fait qu'il n'y a actuellement personne qui joue le rôle de gestionnaire de projet. L'analyste- programmeur peut choisir d'élaborer un plan de projet sommaire de 3 à 5 pages qui présente brièvement l'objectif du projet et la liste des fonctionnalités à implémenter. (Voir Annexe 3).

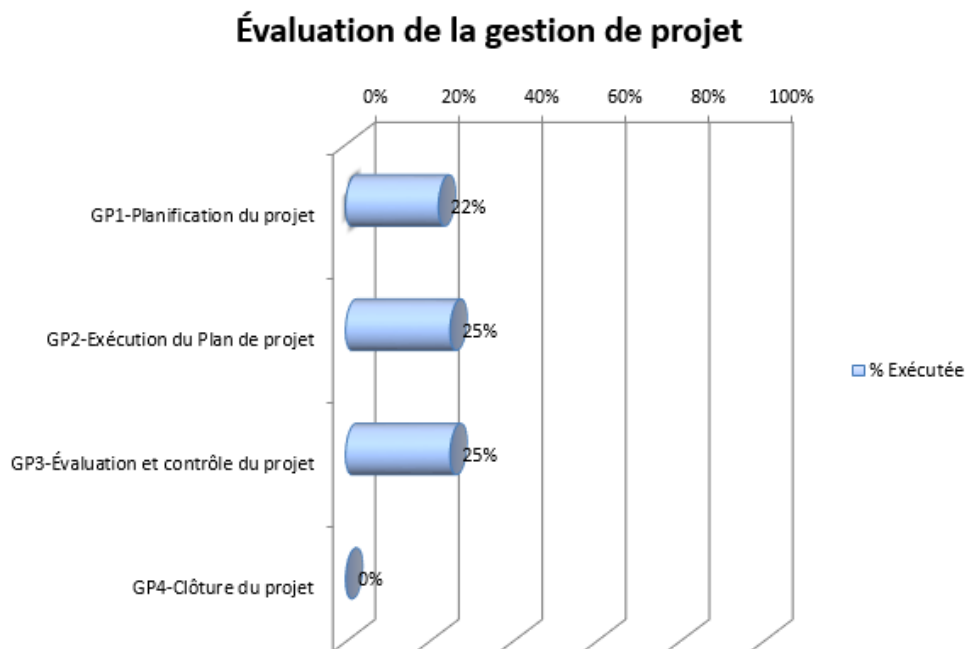


Figure 8 - Évaluation du processus de gestion de projet (PM)

Évaluation de l'implémentation logicielle

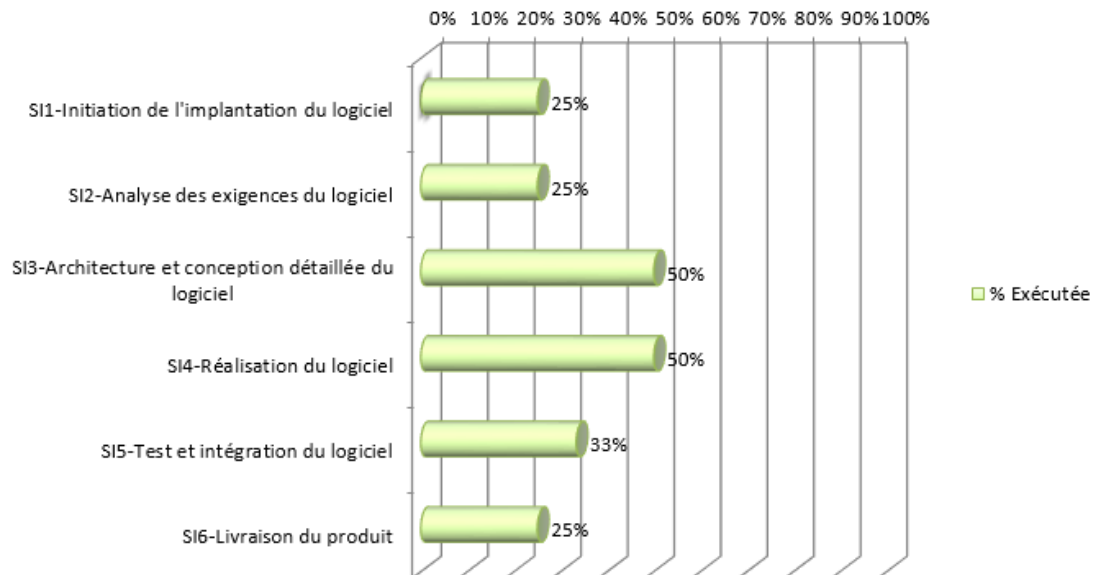


Figure 9 - Évaluation du processus d'implémentation du logiciel

Concernant le processus d'implémentation du logiciel, les détails suivants ont été observés :

- La documentation de la conception est inexistante;
- Aucune utilisation d'outil de gestion de version;
- Présence uniquement de tests fonctionnels (pas de tests unitaires).

Une grille complète des observations détaillées est présentée à l'annexe 4

Conséquences positives

Les conséquences positives qui découlent de ce mode fonctionnement sont : les projets étant de petites tailles, l'exécution de toutes les tâches par une seule personne simplifie le processus et son efficacité, car il n'y a aucun intermédiaire entre le client et le développeur.

Conséquences négatives

- Il y a très peu de traces, tout est dans la tête du programmeur. Si ce programmeur s'en va de l'organisation il sera difficile de maintenir l'application.

CHAPITRE 5

RECOMMANDATIONS

Cette section présente les recommandations émises à la suite de l'évaluation de l'équipe de développement applicatif [23]. Elles sont basées sur les principes de l'agilité [24] et les pratiques recommandées en ingénierie logiciel. Ces recommandations sont regroupées sous trois thèmes: Adoption de l'agilité, les pratiques recommandées du génie logiciel et le soutien organisationnel.

5.1 Adoption de l'agilité

Il serait difficile d'adopter Scrum dans sa totalité sur un projet avec moins de 3 personnes. Cependant, Kanban est moins restrictif dans ce sens. Tout ça pour dire que le choix revient aux développeurs de décider quelle méthode fonctionne mieux dans leur contexte. Toutefois, l'adoption d'une méthode agile serait recommandée pour les avantages énoncés ci-après.

5.1.1 Travailler en équipe

Le principe no. 11 de l'agilité stipule : « *Les meilleures architectures, spécifications et conceptions émergent d'équipes auto organisées* ».

Certains projets développés à la DRI peuvent être implémentés par une seule personne en moins de 3 mois. Toutefois, les projets plus importants, qui mobilisent un développeur à plein temps durant plus de 3 mois ou qui requièrent différentes spécialisations et technologies, devraient être confiés à une équipe.

Les bénéfices attendus sont:

- Favoriser la collaboration, par la répartition des tâches pour être plus efficace et plus rapide;
- Améliorer la qualité par la revue de code entre les pairs;
- Minimiser les risques liés au départ d'un développeur;

- Partager les expériences et les connaissances pour favoriser le développement des compétences;
- Faire émerger des solutions alternatives par la confrontation des avis.

5.1.2 Favoriser l'excellence technique par la formation continue

Le 9^e principe de l'agilité recommande de porter « *une attention continue à l'excellence technique et à une bonne conception* ».

L'un des handicaps observés est que certains membres de l'équipe ne maîtrisent pas les outils et la technologie disponible. L'excellence technique passe obligatoirement par la formation. La direction devrait encourager les développeurs à prendre des formations adaptées à leur besoin et à ceux de l'organisation.

Les bénéfices attendus sont:

- Diminuer les pertes de temps, car la maîtrise des outils permet d'être plus productif;
- Faciliter la maintenance du code par l'adoption des pratiques standards de développement par toute l'équipe.
- Favoriser l'évolutivité et la portabilité du code par l'utilisation des patrons de conception.

5.1.3 Instaurer les réunions de rétrospectives

Le principe de l'amélioration continue dit que: « *À intervalles réguliers, l'équipe réfléchit aux moyens possibles de devenir plus efficace. Puis elle s'adapte et modifie son fonctionnement* ».

Même si la méthodologie Scrum n'est pas retenue, les réunions de rétrospectives sont d'une importance capitale. L'ajout de réunions de rétrospective à une certaine fréquence (une fois/mois par exemple) permettrait aux employés de la DRI de mettre en place des actions visant à améliorer leur efficacité.

Les objectifs de cette pratique sont :

- Capitaliser sur ce qui fonctionne bien pour renforcer l'estime de l'équipe;
- Faire ressortir les problèmes rencontrés et trouver ensemble des solutions;
- Apprendre de ses erreurs.

5.2 Mise en place des pratiques standards

5.2.1 Planification et suivi des projets

Il a été constaté une faiblesse au niveau du processus de gestion de projet. Il serait important d'adopter un outil de gestion de projet et de désigner quelqu'un qui aurait la responsabilité de s'assurer que le processus de gestion de projet soit formalisé.

Les principaux avantages sont de :

- Fournir une vision globale du projet et du volume de travail à réaliser;
- Faciliter la transparence au niveau de l'avancement des travaux;
- Gérer les risques, identifier les problèmes en amont de manière à les corriger rapidement.

5.2.2 Mise en place d'un gestionnaire de code source

L'utilisation d'un logiciel de gestion de version donnerait la possibilité au développeur de revenir en arrière en cas de problème lors du développement d'une fonctionnalité. Les développeurs disposent déjà d'un outil de gestion de code source fourni avec le contrat de licence de Visual Studio, mais l'utilisation de celui-ci n'est pas systématique. La formation des développeurs à cette pratique serait nécessaire au bon déroulement de sa mise en place.

Les bénéfices attendus sont :

- Faciliter la collaboration des développeurs sur un projet

- Gérer plus facilement les différentes versions du code source
- Intégration plus facile d'un nouveau développeur dans l'équipe.

5.2.3 Mise en place des pratiques de tests

Plutôt que d'attendre que le client signale les bogues, avec des commentaires négatifs sur la qualité du produit, il serait bien de mettre en place des jeux de tests qui aident à détecter les défauts. Les outils actuellement utilisés dans l'équipe permettent d'automatiser certains tests. Il est regrettable de ne pas les exploiter.

Les bénéfices attendus sont :

- Assurer la qualité du produit en livrant le logiciel avec le minimum de défaut;
- Augmenter la satisfaction du client;
- Éviter les coûts et les délais supplémentaires dus à la reprise des travaux.

5.2.4 Documenter la conception, le code et la configuration

L'une des valeurs de l'agilité privilégie la livraison un logiciel fonctionnel sans toutefois négliger la valeur de la documentation. La règle de base dans ce domaine est que la documentation devrait être utile, utilisable et utilisée (Trudel, 2011). Il serait bien de documenter tout ce qui serait utile pour faciliter le travail du prochain programmeur qui aura à travailler sur le code.

Les bénéfices de cette pratique sont :

- Faciliter la prise en charge de l'application par les autres développeurs;
- Éviter les erreurs de configuration lors des migrations des applications sur un nouveau serveur;
- Laisser des traces et faciliter la communication.

5.3 Soutien organisationnel

Plusieurs études ont conclu que la réussite d'un projet d'amélioration de processus est conditionnée par la résistance au changement, la culture organisationnelle, la motivation et la volonté des équipes de travail, le soutien inconditionnel de la direction et une communication efficace. Par conséquent, il serait important de prendre en compte les recommandations suivantes.

5.3.1 Gestion du changement

La transition entre l'approche traditionnelle et l'agilité est difficile pour plusieurs raisons, l'une d'elles étant la résistance au changement. Dans ce sens, il serait peut-être utile d'avoir recours aux services d'un coach pour accompagner l'équipe dans le processus.

Les objectifs de cette démarche sont :

- D'aider l'équipe à identifier les bénéfices et les forces de résistance;
- Motiver les membres de l'équipe à maintenir la vision.
- Accompagner les gestionnaires dans le changement de culture organisationnel inhérent au processus.

5.3.2 Appui de la direction

La mise en place des pratiques standard aura certainement des coûts additionnels en termes d'argent, mais aussi de temps. Une baisse de productivité pourrait même être observée dans les premiers instants, le temps que les développeurs maîtrisent les nouveaux outils. Dans ce sens il est important que la Direction voie ça comme un investissement plutôt qu'une dépense.

Il est important que la direction et les membres de l'équipe de développement partagent la même vision et s'engagent ensemble pour favoriser les changements positifs dans l'organisation.

CONCLUSION

Au départ, ce projet était basé sur l'hypothèse que l'application des pratiques du CMMI pourrait aider à optimiser les processus en cours à la division informatique de l'organisation municipale ASL. Cependant au cours de la revue de littérature il est ressorti que ce modèle n'est pas adapté pour les petites organisations. La norme ISO 29110 semble plus approprié puisqu'elle a été conçue spécifiquement pour les très petites organisations.

Les principes de l'agilité et la trousse de déploiement de la norme ISO 29110 ont été utilisés pour évaluer le processus en place à la section « Développement applicatif » de la DRI. Cette évaluation a fait ressortir les faiblesses dans le processus actuel : l'équipe de développement ne suit pas un processus bien défini, chacun fait ce qui lui semble correct. Ce constat vient appuyer l'hypothèse de départ qui suppose que la standardisation des processus pourrait augmenter l'efficacité de l'équipe et la qualité des applications développées.

Les recommandations émises à la suite de l'évaluation se veulent être des jalons qui pourraient guider une démarche d'amélioration continue. La prochaine phase de ce projet aura pour objectif de mettre en place quelques-unes de ces recommandations et de mesurer en fin de compte les bénéfices réels de cette intervention pour l'organisation.

ANNEXE I

Guide d'entrevue

A- Après la présentation sur l'agilité – Question à tous

- Si vous aviez quelque chose à changer hormis votre salaire et votre patron, que changeriez-vous ?
- Au cas où il y aurait beaucoup de changements ici, qu'est-ce vous aimeriez conserver?
- Remplissage du formulaire de Sondage sur l'application des principes agiles

Directeur Service administratif :

- Existe-t-il au niveau de l'organisation des processus définis, par exemple pour l'acquisition d'un progiciel, appel d'offres?
- Existe-t-il un responsable du Contrôle de qualité?
- Dans la politique interne de la Ville est-ce que le service informatique peut être soumis à des audits
- Que pensez-vous de la qualité du travail de l'équipe de la DRI ?
- Comment assurez-vous que l'ensemble des membres de votre équipe partagent la même vision du travail à faire ?

Développeurs (individuel):

- Comment vous organisez-vous pour réaliser les différentes tâches du projet (développement, documentation, test...)
- Quels sont les éléments problématiques dans la façon actuelle de développer des logiciels?
- Si vous aviez quelque chose à changer hormis votre salaire et votre patron, que changeriez-vous ?
- Application des 12 principes agiles dans vos projets?
- Dans l'ensemble de vos projets (passé ou futur) avez-vous identifié un projet qui pourrait être intéressant à développer en mode agile

Développeur principal (Lead):

- Décrivez-nous votre journée de travail typique.
- Si vous aviez quelque chose à changer hormis votre salaire et votre ancien patron, que changeriez-vous ?
- Selon ton expérience, est-ce que ton ancienne équipe devrait passer à l'agilité?
- Quels sont les freins qui pourraient empêcher l'équipe de devenir agile ?
- Si tu devais reprendre un de tes anciens projets lequel aurais-tu aimé développer en mode agile?

Ancien gestionnaire de projet :

- Quelle vision avez-vous de l'agilité ?
- À votre avis quelle est la culture organisationnelle dominante à l'arrondissement: Contrôle, collaboration, compétence ou développement personnel?
- Que pensez-vous de la qualité du logiciel produit à SL?
- Quels sont les éléments positifs qui fonctionnent bien dans votre ancienne équipe?
- Si vous deviez revenir comme chef de division dans votre ancienne équipe qu'est-ce vous auriez changé?
- Selon votre expérience, est-ce que votre ancienne équipe devrait passer à l'agilité?

ANNEXE II

Grille d'évaluation du processus d'Implémentation du logiciel

Ce tableau présente une vue du résultat de l'évaluation du processus d'implémentation du logiciel en utilisant l'outil d'autoévaluation du profil d'entrée de la trousse de déploiement d'ISO 29110.

Activités de processus de développement (SI)	Norme ISO 29110 Partie 5					Votre petite organisation			
		Tâches spécifiques	Intrants	Extrants	Rôles	Est implémentée	Intrants	Extrants	Rôles
SI1-Initiation de l'implantation du logiciel	SI.1.1	Révision du <i>Plan de projet</i> avec les membres de l'équipe afin d'arriver à une compréhension commune et obtenir leur engagement pour le projet.	<i>Plan de projet</i>	<i>Plan de projet [révisé]</i>	PM TL WT	Non			
	SI.1.2	Définir ou mettre à jour l'environnement de l'implantation.	<i>Plan de projet [révisé]</i>		TL WT	Partiel			AN
SI2-Analyse des exigences du logiciel	SI.2.1	Assigner les tâches aux membres de l'équipe conformément avec leurs rôles selon le Plan de projet actuel.	<i>Plan de projet [révisé]</i>		TL WT	Partiel			AN
	SI.2.2	Documenter ou mettre à jour les spécifications d'exigences. -Identifier et consulter les sources d'information (clients, utilisateurs, systèmes antérieurs, documents, etc.) -Analyser les exigences identifiées afin de déterminer la portée et la faisabilité -Générer ou mettre à jour les spécifications d'exigences.	<i>Plan de projet [description du produit]</i>	<i>Spécifications d'exigences</i>	AN CUS	Partiel		<i>Plan de projet</i>	AN
	SI.2.3	Vérification des spécifications d'exigences. -Vérifier l'exactitude et la testabilité des spécifications d'exigences et sa consistance avec la description du produit. Réviser que les exigences sont complètes, sans ambiguïtés et non contradictoire. Les résultats trouvés sont documentés dans les résultats de vérification et les corrections sont faites jusqu'à ce que le document soit approuvé par l'AN. Si des changements significatifs sont requis, une demande de changement est initiée.	<i>Spécifications d'exigences</i> <i>Plan de projet [description du produit]</i>	<i>Résultats de vérification</i> <i>Spécifications d'exigences [vérifiées]</i> <i>Demande de changement [initiée]</i>	AN	Non			
	SI.2.4	Validation des spécifications d'exigences -Valider que les spécifications d'exigences satisfont les besoins et correspondent, incluant l'utilisabilité de l'interface utilisateur. Les résultats trouvés sont documentés dans les résultats de validation et les corrections sont faites jusqu'à ce que le document soit approuvé	<i>Spécifications d'exigences [vérifiées]</i>	<i>Résultats de validation</i> <i>Spécifications d'exigences [validées]</i>	CUS AN	Non			
	SI.2.5	Documenter la version préliminaire du document d'utilisateur du logiciel ou mettre à jour le manuel présent. (optionnel)	<i>Spécifications d'exigences [validées]</i>	<i>Document d'utilisateur du logiciel [préliminaire]</i>	AN	Non			
	SI.2.6	Vérification du document d'utilisateur du	<i>Document</i>	<i>Résultats de</i>	AN	Non			
		Tableau de bord profil d'entrée	Tableau de bord profil basique	Tableau Gestion de projet (PM)		Tableau Implémentation (SI)		Tableau	

ANNEXE III

Gabarit du plan projet actuellement utilisé

Voici la table des matières d'un plan de projet typique rédigé par l'analyste-programmeur à la suite de sa rencontre avec le client. Livrable correspondant à l'étape 2 du processus en cours.

Table des matières

<u>Objectif</u>	4
<u>Portée</u>	4
<u>Fonctionnalités</u>	4
<u>FCT01 - Réservation de salle</u>	4
<u>FCT02 - Annulation de réservation</u>	5
<u>FCT04 - Visionnement des réservations</u>	5
<u>FCT05 – Approuver une demande de réservation</u>	5
<u>FCT06 – Rejeter une demande de réservation</u>	5
<u>FCT07 – Visionnement de l'occupation des salles</u>	5
<u>FCT08 – Modifier les salles</u>	5
<u>FCT09 – Imprimer les horaires des salles</u>	5
<u>Phases</u>	6
<u>Architecture</u>	6
<u>Matrice d'utilisation</u>	6

ANNEXE IV

Grille d'observations

	Observations	Conséquences
Conception		
Activités d'analyse de risques	Les risques sont gérés implicitement par chaque membre de l'équipe, mais il n'y a pas d'analyse réalisée en équipe	Seule une partie des risques du projet est identifiée, mais la potentielle apparition de ceux-ci n'est pas communiquée à l'ensemble des membres de l'équipe
Activités de conception	Les membres de l'équipe étant seuls sur leurs projets, ils font leur propre activité de conception et décident eux-mêmes de la conception du produit.	L'architecture n'est pas toujours optimale, et varie d'un projet à l'autre.
Réalisation		
Pratique de tests	Présence de tests fonctionnels uniquement	La modification des fonctionnalités par un autre développeur risque d'introduire des défauts. L'absence d'autres tests engendre un risque de défauts plus élevé au sein du code source et ne fournit pas une assurance qualité optimale du code au moment de sa livraison.
Documentation		
Erreurs/Obstacles rencontrés	Certains développeurs ont un logbook dans lequel ils gardent des traces pour leur dossier personnel.	L'information ne pourra pas être retrouvée facilement.
Retour d'expérience	Des activités de retour d'expérience ne sont actuellement pas effectuées	L'amélioration continue n'est pas encouragée
État d'avancement du projet	Il n'y a pas de document mis en place afin de mesurer l'état d'avancement des projets. Certains développeurs ont testé le tableau Kanban pendant un certain temps, mais la pratique n'a pas été poursuivie, car les bénéfices n'ont pas été identifiés.	Il est difficile pour la direction de faire un suivi de l'avancement du projet ainsi que des coûts et des délais en vue d'analyser les problèmes rencontrés et de les corriger.
Interactions		

Présentation des résultats de la phase de développement au client	Le client est rencontré aussi souvent que nécessaire pour valider ou tester	Conséquences positives: Le client est à proximité ce qui donne à l'équipe la possibilité d'avoir des retours sur l'application et s'assurer de bien livrer ce qu'il demande.
Communication entre les membres de l'équipe	Tous les membres de l'équipe sont dans le même espace. Ils travaillent sur des projets différents, mais ils se parlent.	Conséquences positives: ils s'entraident à trouver des solutions à de possibles problèmes
Résistance au changement	Les membres de l'équipe sont plutôt enclins à transiter vers l'agilité à condition que cela leur apporte une valeur ajoutée sur leurs projets. Une personne paraît néanmoins réticente	Le virage à l'agilité ne pourra pas se faire pour tous les projets
Organisation		
Disposition du mobilier dans l'espace de travail	Plutôt favorable, avec une salle de réunion, un tableau blanc et tout le monde dans la même pièce. Mais les cubicules créent un certain isolement.	
Processus documentés et visibles par l'ensemble des membres concernés dans l'organisation	Pratiques communes, mais pas de processus écrit	Un nouvel employé aura du mal à se retrouver et devra toujours demander aux autres de comment procéder.
Gestion des demandes de modification	les développeurs prennent eux-mêmes la décision concernant les demandes de modifications du client.	
Outils utilisés		
Recueil des exigences du client	Gardé dans le répertoire du projet sous forme d'un fichier Word ou Excel	

Suivi des tâches	Certains développeurs utilisent des « post-it » pour suivre l'évolution de leurs projets L'utilisation n'est pas systématique	L'évolution des projets reste obscure pour toute autre personne
Outil de gestion de version	Outil Azure DevOps possédé, mais non utilisé par les membres de l'équipe par manque de connaissances à ce sujet	La gestion des versions est problématique quand 2 développeurs doivent travailler sur le même code.

BIBLIOGRAPHIE

1. Baud, J.-L. (2016) ITIL v3 Comprendre la démarche et adopter les bonnes pratiques, France, éditions ENI.
2. Dufay, F. (2010) CMMI par l'exemple : Pour une mise en place opérationnelle, France, Eyrolles éditeur.
3. Basque, R. CMMI 1.3 : Guide complet de CMMI-DEV et traduction de toutes les pratiques CMMI-ACQ et CMMI-SVC, Dunod éditeur, 296 p.
4. Coallier F. et Tremblay D., Notes de cours MTI825 La gestion des services TI, Montréal, ÉTS, Hiver 2020.
5. April, A. et Abran, A (2016) Améliorer la maintenance du logiciel, Loze-Dion éditeur, 384 p.
6. Boisvert, M. et Trudel, S. (2011) Choisir l'agilité : Du développement logiciel à la gouvernance, Montréal, Canada, Dunod éditeur, 320 p.
7. Marchand, S. (2018) Évaluation d'une mise en œuvre d'une approche agile dans une organisation municipale, Montréal, UQAM.
8. O'Connor R.V. et Laporte C.Y. Implementing Process Improvement in Very Small Enterprises with ISO/IEC 29110 - A Multiple Case Study Analysis, proceedings of the 10th international conference on the Quality of Information and Communications Technology (QUATIC), 6-9 , Lisbon, Portugal, pp: 125-130, (2016).
9. Muñoz M., Mejia J., et Laporte C.Y. (2018) Reinforcing Very Small Entities Using Agile Methodologies with the ISO/IEC 29110, proceedings of the international conference on software process improvement, pp: 88-98.
10. Coque-Villegas, S., Jurado-Vite, V. et coll. (2018) Analysis of experiences of improvement of software development processes in SMEs. Universidad Politécnica Salesiana, Equateur, UNEMI, 2018
11. Site web du prof Laporte C. <http://profs.etsmtl.ca/claporte/English/VSE/index.html>
12. Calvo-Manzano J. A., L.-M., et coll. (2015) How small and medium enterprises can begin their implementation of ITIL?, revista facultad de Ingenieria, Universidad de Antioquia, no.77: 127-136.
13. Galvana, S. Morab, M. et coll. (2015) A Compliance Analysis of Agile Methodologies with the ISO/IEC 29110 Project Management Process, science direct, vol. 64:188-195.
14. Vishal V. (2019) ITIL4 and Cobit, a white paper, Axelos [En ligne] <https://www.axelos.com/case-studies-and-white-papers/itil-4-and-cobit-white-paper>
15. Dufour, C. et Laporte, C.Y. (2014) Implantation de la norme ISO/IEC 20000 en gestion des services TI dans un petit service informatique, [En ligne] https://www.researchgate.net/publication/261082923_Implantation_de_la_norme_ISOIEC_20000_en_gestion_des_services_TI_dans_un_petit_service_informatique

16. Kabli, S. Conduite d'une démarche vers une certification ISO 20000 : « Trousses de déploiement », ÉTS, 2009, 35 pages.
17. Jeljeli, H. et Laporte, C.Y. Mise en œuvre de processus logiciels de l'ISO/CEI 29110 dans une entreprise en démarrage, ÉTS, pp. 34-41.
18. Lebel, K. et Laporte, C.Y. (2018) Développement, en mode Agile, d'une application à l'aide de la norme ISO/CEI 29110 au sein d'une grande société d'État québécoise, ÉTS, pp. 48-58.
19. Bustard, D., Wilkie, G., Greer, D. (2013) Towards optimal software engineering: learning from agile practice, *Innovations Syst Softw Eng*, no. 9:191-200. [En ligne] <https://doi.org/10.1007/s11334-013-0216-9>
20. Moreta, L.M.L. (2015) ITIL in Small to Medium-Sized Enterprises: Toward a Proposal Based on an ITIL Processes Implementation Sequence and a Profile Scheme Strategy for Implementing the First Process in the Sequence, master thesis, Universidad Politecnica de Madrid, Spain, UPM, 118p.
21. ISO 29110-5-3 (2018) Iso/iec tr 29110-5-3:2018 Systems and software engineering – lifecycle profiles for very small entities (VSEs) – part 5-3 service delivery guidelines, [En ligne] <https://www.iso.org/obp/ui/#iso:std:iso-iec:tr:29110:-5-3:ed-1:v1:en>
22. Methawachananont, A. Buranarach, M. et coll. (2020) Software Process Capability Self-Assessment Support System Based on Task and Work Product Characteristics: A Case Study of ISO/IEC 29110 Standard, *IEICE TRANS*.
23. Dassonville A, Benzaki C. Norceide M., Évaluation de l'agilité à la Division des ressources informationnelles de la Ville du Bonheur, UQAM, 2019.
24. AgileAlliance. Manifeste pour le développement agile de logiciels, 2001.
25. Version One, 13th Annual State of Agile Report, 2019